


# Controlling

## Content Dynamically

---

### Objectives

- ▶ Understand dynamic content
- ▶ Insert content dynamically
- ▶ Delete content dynamically
- ▶ Modify content dynamically
- ▶ Incorporate an advanced content function
- ▶ Replace graphics dynamically
- ▶ Bind data
- ▶ Manipulate bound data dynamically

Just as dynamic HTML (DHTML) allows you to create pages whose style changes instantly based on user actions, it also provides tools that allow users to immediately modify a page's content. You can use this feature, known as **dynamic content**, to generate all or part of the page when it is opened, or even to alter the page's contents in response to user events.  The manager of Nomad Ltd's retail division has heard about dynamic HTML and has asked Lydia to add dynamic content features to some of their Web pages to increase their interactivity. Lydia plans to use dynamic content features that will allow users to adapt the pages to their needs.



# Understanding Dynamic Content

Dynamic HTML includes many tools for altering a Web page's appearance in response to user actions. Using scripts to change text attributes such as color and font size alter the style of elements, leaving the elements themselves, such as text or images, unchanged. Dynamic content tools, however, allow your Web page elements to move or change based on user input. These changes can include the elements themselves as well as the HTML tags associated with elements. Dynamic content can create an effect similar to an expanding outline. The outline actually uses a style attribute, "display" or "appearance," to simply show or hide text while the text remains part of the Web page. True dynamic content involves element reordering and replacing. As she learns about dynamic content, Lydia identifies several of its main uses and thinks about ways she can use it on the Nomad Ltd Web site.

## Details



### Pointing

Dynamic content allows you to change an element in response to a user's mouse pointer movements. You already have learned about the formatting changes you can create using dynamic style. Now, using dynamic content, you can make your page's text and graphic contents available to user changes. Figure K-1 shows a Web page displaying an alternate graphic in response to user pointing.



### Run-time activities

Dynamic content tools can create portions of your Web pages for you at **run time**, the period when a browser first interprets and displays the Web page and runs scripts. A simple case would be a script that displays the text "Good Morning!" or "Good Evening!" based on the time of day according to your computer's clock. You also can program a page to generate a table of contents for the page at run time, which allows you to change the page's structure and contents without also revising the TOC each time you make a change.



### HTML tables

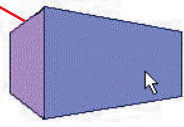
In addition to standard tools for working with Web page text, dynamic content includes special features for easily creating and working with tables. You can use dynamic content tools to associate an external database with a Web page, a process known as **data binding**. Data binding allows the user's browser to generate a Web page table from an external data file at run time. By adding some lines of script, you also can allow users to sort the table right on your Web page. Figure K-2 shows a dynamically generated table in a Web page that has been sorted by the Web page user.



**FIGURE K-1:** Dynamic content responding to user pointing

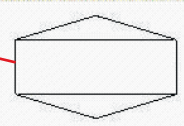
As you narrow your choices, click the Remove button for each tent that you're no longer considering, to remove it from the page.

## Tent footprints and descriptions




**XTC Starlite**  
One of the lightest, most compact three-season tents available. Featuring two-pole clip design with a built-in vestibule.

[Remove Starlite](#)



**Amano Brevifolia**  
The simple, vaulted design characterized by two doors and two vestibules returns with the 2000 Brevifolia model. New features include: ground level, rainfly with vents, and vaulted sleeves for smoother pole feeding.

[Remove Brevifolia](#)



**Amano Trifolia**

Color graphic replaces original line art in response to pointer

Line art of tent design

**FIGURE K-2:** Table sorted by user

Tent	Catalog number	Area (sq ft)	Vestibule (sq ft)	Description	Capacity	Weight	Price
XTC Starlite	BR-370	34	10	Staked	1 person	4 lbs. 3 oz.	\$150
Amano Brevifolia	BT-356	38.5	19.6	Freestanding	2 people	5 lbs. 8 oz.	\$215
Amano Trifolia	BT-358	49	25.7	Freestanding	2 people	7 lbs.	\$250
Vista Hillside	BZ-339	32	15.3	Staked	1 person	4 lbs.	\$120
Vista Hilltop	BZ-367	37.5	19.5	Staked	1 person	5 lbs. 3 oz.	\$160
Vista Peak	BZ-323	42.5	24.4	Freestanding	2 people	6 lbs. 3 oz.	\$210
Vista Summit	BZ-334	51.5	28	Freestanding	2 people	7 lbs. 10 oz.	\$275

Bound data not sorted

Tent	Catalog number	Area (sq ft)	Vestibule (sq ft)	Description	Capacity	Weight	Price
XTC Starlite	BR-370	34	10	Staked	1 person	4 lbs. 3 oz.	\$150
Vista Hillside	BZ-339	32	15.3	Staked	1 person	4 lbs.	\$120
Vista Hilltop	BZ-367	37.5	19.5	Staked	1 person	5 lbs. 3 oz.	\$160
Amano Brevifolia	BT-356	38.5	19.6	Freestanding	2 people	5 lbs. 8 oz.	\$215
Vista Peak	BZ-323	42.5	24.4	Freestanding	2 people	6 lbs. 3 oz.	\$210
Amano Trifolia	BT-358	49	25.7	Freestanding	2 people	7 lbs.	\$250
Vista Summit	BZ-334	51.5	28	Freestanding	2 people	7 lbs. 10 oz.	\$275

Table sorted in response to click on column head

For more information on Nomad Ltd outdoor supplies, please email our [sales department](#)



## Dynamic HTML features are not discrete

Although you can divide dynamic HTML effects into categories, such as dynamic style and dynamic content, the tools you use to create these effects often overlap. For example, to implement cross-browser dynamic style, you often need to identify the brand of the user's browser and then add lines to the embedded

style sheet that are appropriate for the browser. Because you are adding code to the Web page at run time, this is a use of dynamic content to create dynamic style! As you learn more DHTML features and tools, their implementation will overlap increasingly.



# Inserting Content Dynamically

Adding content at run time with scripts can allow you to create impressively customized and versatile Web pages. Because the DOM provides access to all the elements of a Web page, you can use scripts to alter any page elements based on conditions on the user's computer or on the page's current contents. Lydia's first project for the retail department is a Web page that compares the tents that Nomad Ltd sells. She wants to add a statement announcing the number of tent models that users can read about on the page. She can use a script to count the number of tent descriptions on the page and then insert the number dynamically in the page header statement that appears at the bottom of the page when the page loads. This means that the page header statement will still show the correct number even after the sales department adds to or removes tents and their descriptions from its tent selection.

## Steps 1234

1. Start your Web browser program and cancel any dial-up activities, then open the file **HTML K-1.htm**

The page shows each tent's floor plan, or footprint, along with the tent's description.

2. Start your text editor program, open the file **HTML K-1.htm**, then save it as text document with the filename **Tent count.htm**

Lydia has included a function in the page header that counts the number of tent-description headings in the page and assigns the number to the variable `totalTents`.

3. Scroll to the bottom of the page code, highlight the text **[replace with tent count code]**, then press **[Delete]**

### QuickTip

Be sure to type a space after the word *describes* and a space before the word *tent*.

4. Type the following code, pressing **[Enter]** at the end of each line

```
<SCRIPT>
<!--
if (IE4) {
    countHeaders()
    document.write("<H1 ALIGN='center'>This page describes ")
    document.write(totalTents)
    document.write(" tent models.</H1>")
}
//-->
</SCRIPT>
```

Figure K-3 shows the completed Web page code containing the script. The code formats the text "This page describes" and "tent models." as centered on the page with an H1 format. Between the two bits of text, the script uses the `document.write` method to insert the value counted by the `countHeaders` function, which is assigned to the variable "totalTents." Because the script that counts the headers works only in Internet Explorer 4, the script begins by checking the browser version.

5. Check your document for errors, make changes as necessary, then save **Tent count.htm** as a text document

6. Open **Tent count.htm** in your Web browser, then scroll to the bottom of the page

Figure K-4 shows the Web page in Internet Explorer 4. The H1 text Lydia added appears near the bottom of the page. The statement includes the number of tents counted by the `countHeaders` function and inserted with a script.

FIGURE K-3: Completed Web page code

Text and  
script for  
tent count  
statement

```
<DIV ID="tent7" name="tent">

<DIV CLASS="tenthead"><IMG SRC="summit.jpg" ALIGN="left">Vista Summit</DIV>

<DIV>Comfortable, rugged, 4-season tent. Quick setup, full rainfly, integral
vestibule, large door. Factory sealed, mesh window and door for
ventilation.</DIV><BR><BR>

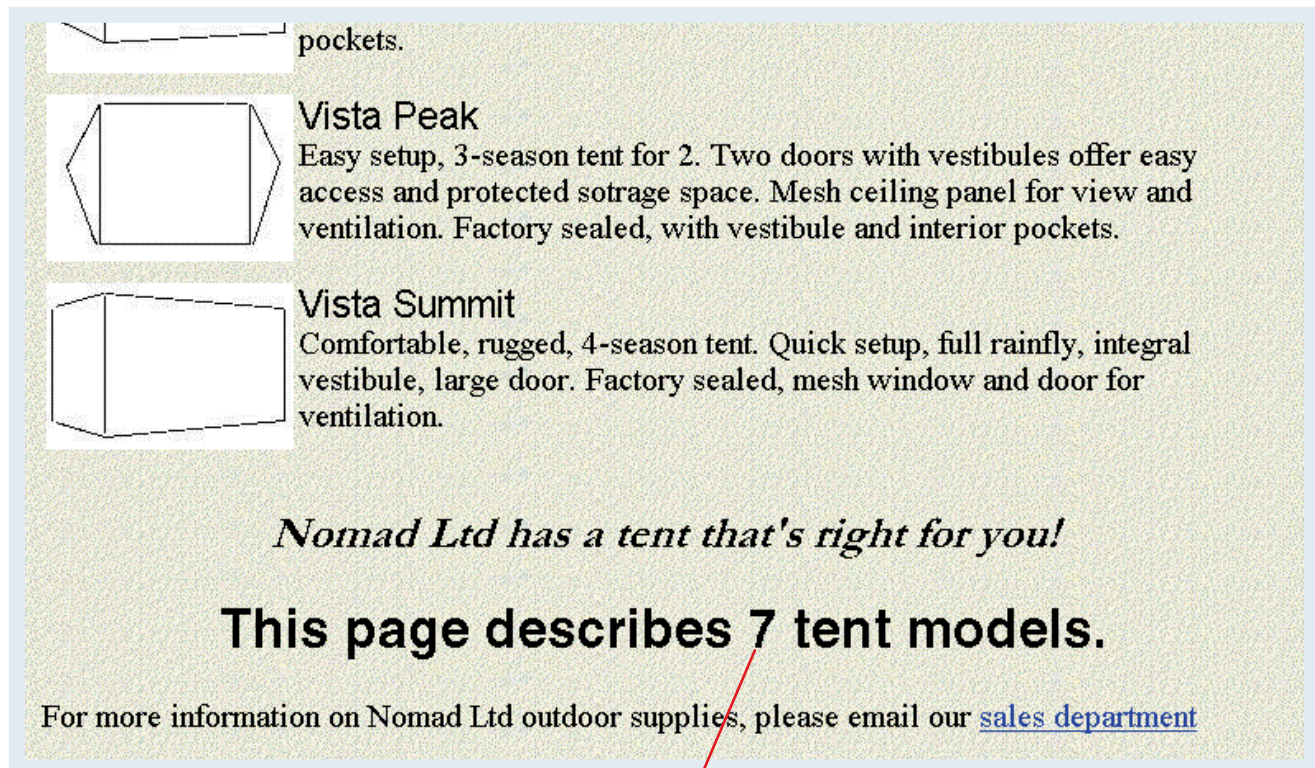
<H2 ALIGN="center">Nomad Ltd has a tent that's right for you!</H2>

<SCRIPT>
<!--
if (IE4) {
    countHeaders()
    document.write("<H1 ALIGN='center'>This page describes ")
    document.write(totalTents)
    document.write(" tent models.</H1>")
}
//-->
</SCRIPT>

<DIV>For more information on Nomad Ltd outdoor supplies, please email our <A
HREF="MAILTO:sales@nomadltd.com">sales department</A></DIV>


</BODY>
</HTML>
```

FIGURE K-4: Web page displaying tent count



Total calculated  
by counting script

# Deleting Content Dynamically

In addition to adding Web page elements dynamically at run time, you can script your Web page to allow users to tailor it to suit their needs. For example, some scripts can allow users to delete elements from a Web page, including text and graphics. This feature—especially useful in a content-laden page—allows the user to pare down the content in order to view only pertinent elements or sections.  Because users of the tent comparison page will be trying to select a tent based on their needs, Lydia thinks it would be helpful to allow users to remove information they are not interested in from the page for tents.

## Steps 1234

1. Open the file **HTML K-2.htm** in your text editor, then save it as a text document with the filename **Tent delete.htm**
2. Scroll down the page to view the body text describing the first tent, the XTC Starlite, select the text **[insert button code for tent1]**, then press **[Delete]**
3. Type the following code, pressing **[Enter]** at the end of each line

```
<SCRIPT LANGUAGE="javascript">
<!--
if (IE4) {
```

### Trouble?

To specify the null value, be sure to type single quotes after HTML=.

4. Press **[Tab]**, then type **document.write("<BUTTON CLASS='button' onClick=tent1.outerHTML=">Remove Starlite</BUTTON>")** and press **[Enter]**

The **<BUTTON>** tag set creates a button with a customized function in Internet Explorer 4 only. The text between the tags is the label that appears on the button. Lydia has inserted a class definition called **.button** in the page's embedded style sheet. She uses the **onClick** event handler to change the **outerHTML** property of the object named **tent1**, which includes the description and graphic for the first tent. An element's **outerHTML** property includes the element contents and the tags surrounding it, so changing the property to a null value removes the element and its surrounding tags from the Web page.

5. Type **}** and press **[Enter]**, then type the following closing script tags, pressing **[Enter]** at the end of each line

```
//-->
</SCRIPT>
```

6. Repeat Steps 2 through 5 for the remaining six tent descriptions, substituting the button object names and tent names as listed in Table K-1

Figure K-5 shows the Web page containing the button code for the first two tent descriptions.

7. Check the document for errors, make changes as necessary, then save **Tent delete.htm** as a text document

8. Open **Tent delete.htm** in your Web browser and scroll down the page until the Amano Brevifolia description appears in the document window

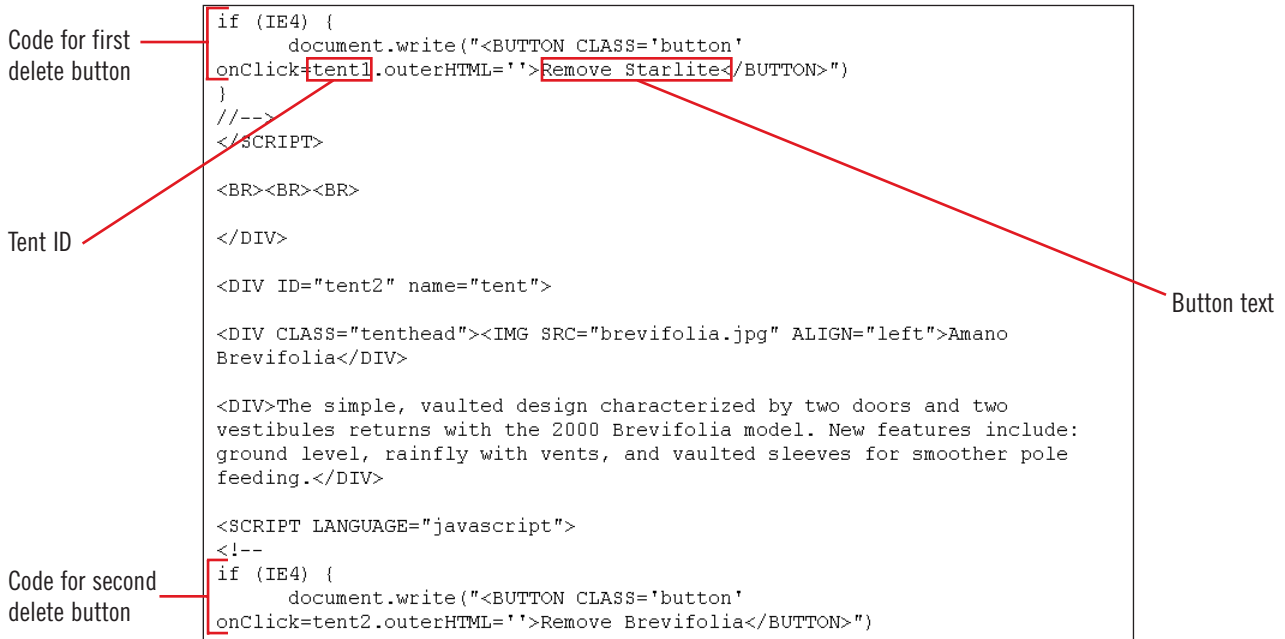
Internet Explorer 4 displays the "Remove Brevifolia" button, but other browsers do not show the buttons. Even though the function for deleting content only works in Internet Explorer 4, your cross-browser Web page still displays the basic tent information in other browsers without causing JavaScript errors.

9. If you are using Internet Explorer, click the **Remove Brevifolia button**

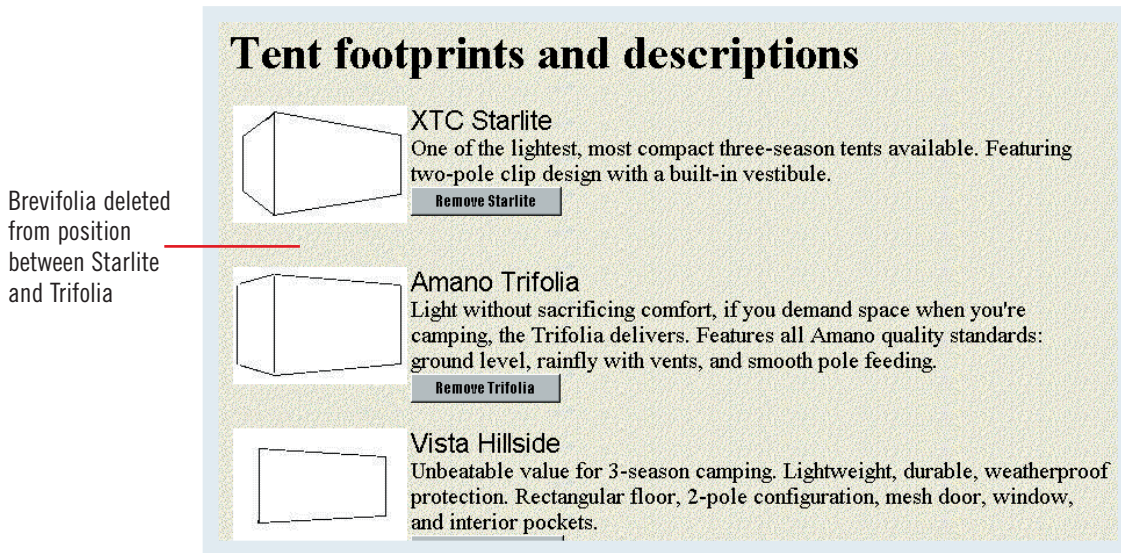
As Figure K-6 shows, the Web browser removes the tent's description and graphic. Next, Lydia will need to be sure the counter reflects this change by updating the number of tent descriptions displayed.



**FIGURE K-5: Web page containing code for delete buttons**




**FIGURE K-6: Web page with Brevifolia removed**



**TABLE K-1: Tent description IDs and button text**

description number	substitute for "tent1"	substitute for "Starlite"
2	tent2	Brevifolia
3	tent3	Trifolia
4	tent4	Hillside
5	tent5	Hilltop
6	tent6	Peak
7	tent7	Summit

# Modifying Content Dynamically

Dynamic content doesn't stop at adding or deleting static Web page content. Also, you can create pages that allow their contents to change in response to various events. You can use this feature to create a basic useful function, such as a DHTML clock, as part of a Web page. A DHTML clock function changes the contents of a text element displaying the time (for example, once per second) in response to the passing of time. You also can add interactivity by modifying page content in response to user actions.  Because her page allows users to remove descriptions for tents that don't fit their needs, Lydia wants to ensure that the statement showing the number of tents available displays the correct number after user deletions.

## Steps 1 2 3 4

1. Open the file **HTML K-3.htm** in your text editor, save it as a text document with the filename **Tent update.htm**, then scroll down the page until the function **reCount** appears in the document window

Notice that Lydia has added the function named **reCount**. The function **reCount** subtracts 1 from the total count of tent descriptions on the page and then uses the **innerHTML** property to update the number that appears in the statement at the bottom of the page. **innerHTML** replaces an element but leaves its enclosing HTML tags intact. Lydia uses **innerHTML** because she wants to replace only the number, which is within HTML tags, and not any of the surrounding text or HTML tags. Lydia has written the code so that each of the buttons that removes a tent description from the page triggers the **reCount** function.

2. Scroll down the page until the opening **<BUTTON>** tag for **tent1** appears  
Notice that Lydia has added a reference to the **reCount** function in the **onClick** event handler. She has added this reference for each of the buttons.

3. Scroll to the bottom of the Web page code, select the text **[replace with code to write opening SPAN tag]**, then press **[Delete]**

4. Type **document.write("<SPAN ID='textnum'<>")**

5. Select the text **[replace with code to write closing SPAN tag]**, press **[Delete]**, then type **document.write("</SPAN>")**

Figure K-7 shows the completed code containing the SPAN tags. By inserting the SPAN tags with an ID value, you create an inline object named "textnum" that you can manipulate with scripts. Lydia's **reCount** function changes **textnum**'s **innerHTML** property each time the user clicks one of the delete buttons. This use of dynamic content keeps the contents of the Web page statement current with page changes produced by user actions.

6. Check your document for errors, make changes as necessary, then save **Tent update.htm** as a text document

7. Open **Tent update.htm** in your Web browser, then scroll to the bottom of the page  
In Internet Explorer 4, notice that the tent description total, which is currently 7, displays in the statement.

8. If you are using Internet Explorer, click the **Remove Summit button**  
The browser removes the description for the Vista Summit tent. Simultaneously, it updates the tent description total to 6, as Figure K-8 shows.

9. If you are using Internet Explorer, click the **Remove Hilltop button**  
The browser removes the Vista Hilltop description, and again changes the tent total to reflect the current number of descriptions on the page.



FIGURE K-7: Code containing SPAN tags

JavaScript to write opening and closing SPAN tags inserted

```
</SCRIPT>

<BR><BR>

</DIV>

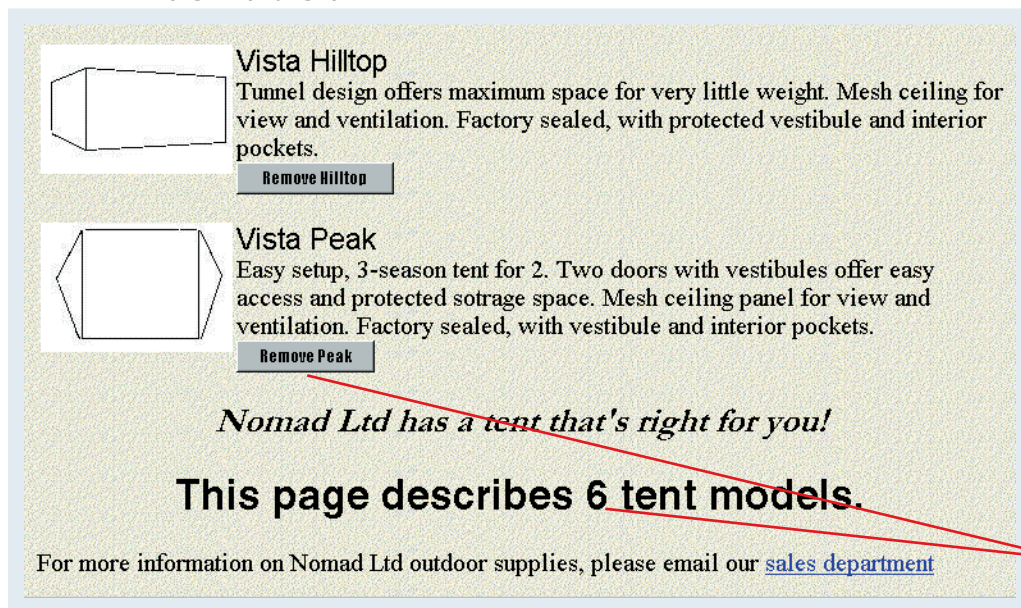
<H2 ALIGN="center">Nomad Ltd has a tent that's right for you!</H2>

<SCRIPT>
<!--
if (IE4) {
    countHeaders()
    document.write("<H1 ALIGN='center'>This page describes ")
    document.write("<SPAN ID='textnum'>")
    document.write(totalTents)
    document.write("</SPAN>")
    document.write(" tent models.</H1>")
}
//-->
</SCRIPT>

<DIV>For more information on Nomad Ltd outdoor supplies, please email our <A
HREF="MAILTO:sales@nomadltd.com">sales department</A></DIV>

</BODY>
</HTML>
```

FIGURE K-8: Web page displaying updated total



Tent total updated to 6 because Summit tent description deleted



## Tool Tips and other floating help

In both Internet Explorer and Netscape Navigator, you can create floating windows that display text relevant to an element when the user moves the cursor over it. This effect is similar to ToolTips in Microsoft applications. These windows are dynamic modifications of the page content in response to user actions. For images, you can use the ALT property to specify the text that displays in a floating window when the

user holds the mouse pointer over the image. For other Web page elements, Netscape Navigator versions 3 and 4 require a script to add this effect. However, you can add this effect in Internet Explorer 4 by adding TITLE="text" to the opening tag for the element. Because these floating windows add and remove Web page text, they are part of your set of dynamic content tools.

# Incorporating an Advanced Content Function

Combining different DHTML tools in your scripts allows for a great variety of possible new dynamic content effects, including different ways of presenting or changing your page elements. You can make your Web page unique as well as make it easier for users to read and navigate by incorporating special features into your Web page. These features also can increase your Web page readership. Lydia sees a page on the Web containing a script that cycles through different Web page elements in the same spot. This effect is like a slide show, with each new segment of text appearing after a short interval. She decides to use this feature on the tent page she is developing to display some additional information about Nomad Ltd's products.

## Steps 1234

1. Open the file **HTML K-4.htm** in your text editor, save it as a text document with the filename **Tent cycle.htm**, then scroll down the page until the code for the function cycle appears in the document window

Notice that Lydia entered the function cycle in the page head script. This function replaces an object's contents at regular intervals by using the innerHTML property in conjunction with the script for counting time.

2. Scroll down the Web page code until the top of the body section appears in the document window, select the text **[replace with text cycle script]**, then press **[Delete]**
3. Type the following script, pressing **[Enter]** at the end of each line:

```
<SCRIPT LANGUAGE="javascript">
<!--
function addCycle() {
```

4. Press **[Tab]**, the type **cycle(txt1, "Hiking,Bicycling,Camping,Kayaking,Climbing, find all your gear at,nomadltd.com", 30)** and press **[Enter]**

This line defines the display parameters for the text you want to cycle as follows: txt1 indicates the name of the object whose value will be cycled; the text in quotes separated by commas specifies the different words and phrases that should cycle; and the number 30 tells how long one word or phrase should display before cycling to the next word or phrase.

5. Type the remaining code, pressing **[Enter]** at the end of each line

```
}
if (IE4) {window.onload = new Function("addCycle()")}
//-->
</SCRIPT>
```

Figure K-9 shows the Web page source code containing the completed script. This script triggers the function cycle, which begins to cycle text after the page loads.

6. Check your document for errors, make changes as necessary, then save Tent cycle.htm as a text document and open it in your Web browser

In Internet Explorer 4, the cycling text appears in the top right corner of the page, as shown in Figure K-10. The text cycles at a regular 3-second interval as specified by 30 in the script.



**FIGURE K-9: Page source containing script to call cycle function**

Script to  
invoke text  
cycling  
function

```
<IMG SRC="nomad.jpg" ALIGN="left">

<DIV ID="txt1" ALIGN="right" CLASS="tenthead" STYLE="font-size: 18pt"></DIV>

<SCRIPT LANGUAGE="javascript">
<!--
function addCycle() {
    cycle(txt1, "Hiking,Bicycling,Camping,Kayaking,Climbing,find all your
gear at,nomadltd.com", 30)
}

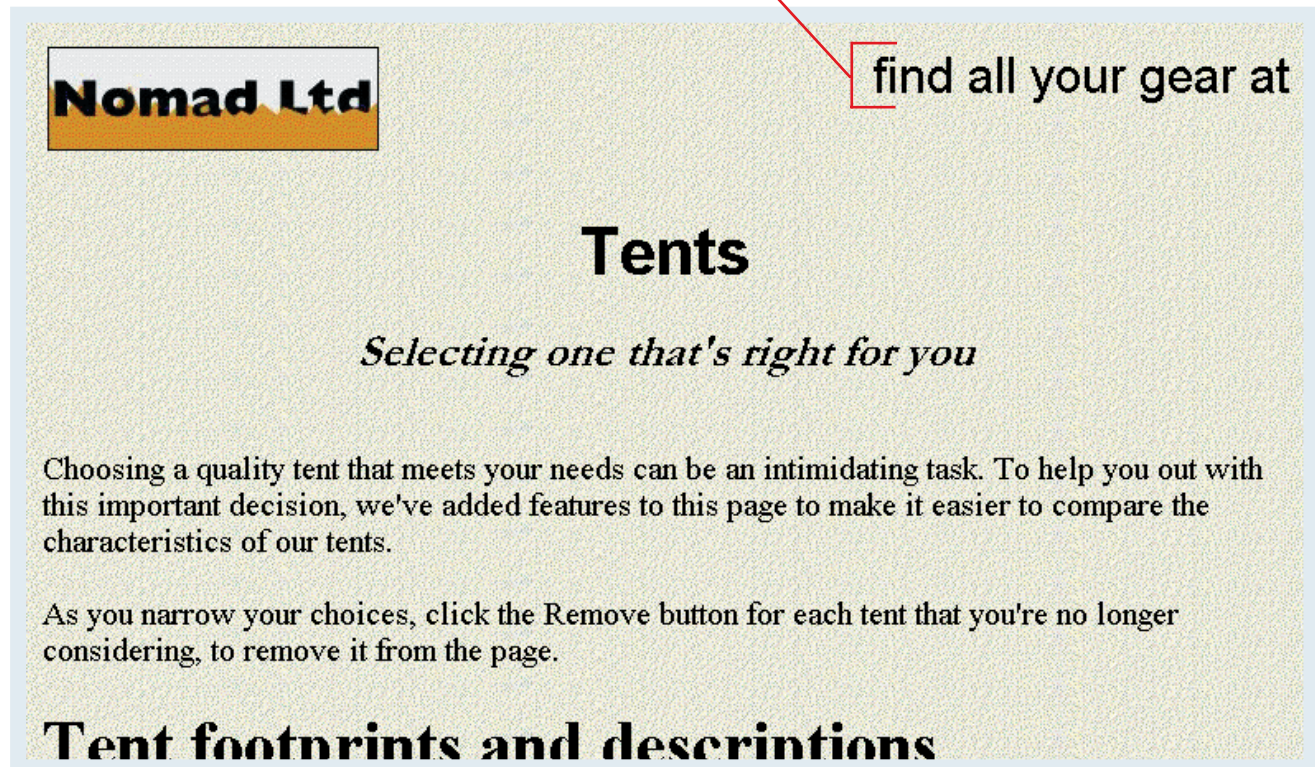
if(IE4) {window.onload = new Function("addCycle()")}
//-->
</SCRIPT>

<BR><BR><BR>
<DIV ALIGN="center" STYLE="font-size: 24pt; font-weight: bold; font-family:
arial; font-style: normal">Tents</DIV>
<H2 ALIGN="center">Selecting one that's right for you</H2>
<BR>

<DIV>Choosing a quality tent that meets your needs can be an intimidating
task. To help you out with this important decision, we've added features to
this page to make it easier to compare the characteristics of our tents.
<BR><BR>
As you narrow your choices, click the Remove button for each tent that
```

Text content changes  
every few seconds

**FIGURE K-10: Web page displaying cycling text**





# Replacing Graphics Dynamically

All the examples so far have used dynamic content tools to modify a Web page's text, but these features are equally valid for other page elements, including graphics. In a simple scenario, you can use dynamic content features to change the graphic displayed using the `onMouseOver` event handler. In a more complex scenario, you could gradually change a graphic's size to create the effect of animation. Lydia wants to use color to highlight the element that the user is currently pointing to. However, rather than using dynamic style, she creates colored versions of each of the tent footprint graphics. The color version of a text footprint graphic will appear in response to mouse movement over each graphic or its associated text.

## Steps 1234

### QuickTip

Remember to type an apostrophe followed by a quotation mark, not three apostrophes.

1. Open the file **HTML K-5.htm** in your text editor, then save it as a text document with the filename **Tent color.htm**
2. Scroll down the Web page code until `<DIV CLASS="tenthead">` appears in the document window, select the text **[replace with star event handlers]**, then press **[Delete]**
3. Type `onMouseOver="star.src='starcolor.jpg'" onMouseOut="star.src='starlite.jpg'"`
4. Scroll down the Web page code, select the text **[replace with brev event handlers]**, press **[Delete]**, then type `onMouseOver="brev.src='brevcolor.jpg'" onMouseOut="brev.src='brevifolia.jpg'"`  
 Figure K-11 shows the completed code for the first two tent items. Notice that the `IMG` tag for each tent has a unique ID attribute. The `onMouseOver` event swaps a color graphic of the tent floorplan for the original image source. The `onMouseOut` event replaces the color image with the original black and white graphic.
5. Repeat Step 4 for the remaining five list items, using the IDs and graphic files listed in Table K-2
6. Check your document for errors, make changes as necessary, then save **Tent color.htm** as a text document
7. Open **Tent color.htm** in your browser
8. Scroll down to the list of tent descriptions, then move your mouse pointer over the heading or graphic for the XTC Starlite  
 See Figure K-12. When you move the cursor over the black and white outline or its associated heading in Internet Explorer 4, the image is replaced with a color graphic. Even though you are simply swapping one graphic for another, this action creates the illusion of modifying the original graphic, much like changing text color using style sheets.
9. Move the mouse pointer off the first list item  
 The graphic changes back to the black and white version. Notice that if you move the mouse pointer over other tent graphics, they change color in response to the mouse movement.

FIGURE K-11: Event handlers for first and second list items

Event handlers for Starlite inserted in DIV tag

ID attribute

Event handlers for Brevifolia inserted in DIV tag

IMG source

```
<DIV ID="tent1" name="tent">
  <DIV CLASS="tenthead" onMouseOver="star.src='starcolor.jpg'"
    onMouseOut="star.src='starlite.jpg'"><IMG SRC="starlite.jpg" ALIGN="left"
    ID="star">XTC Starlite</DIV>
  <DIV>One of the lightest, most compact three-season tents available.
    Featuring two-pole clip design with a built-in vestibule.</DIV>

  <SCRIPT LANGUAGE="javascript">
    <!--
    if (IE4) {
      document.write("<BUTTON CLASS='button'
        onClick=tent1.outerHTML='',reCount()>Remove Starlite</BUTTON>")
    }
    //-->
  </SCRIPT>

  <BR><BR><BR>
</DIV>

<DIV ID="tent2" name="tent">
  <DIV CLASS="tenthead" onMouseOver="brev.src='brevcolor.jpg'"
    onMouseOut="brev.src='brevifolia.jpg'"><IMG SRC="brevifolia.jpg"
    ALIGN="left" ID="brev">Amano Brevifolia</DIV>
```

FIGURE K-12: Web page showing substituted graphic

Color graphic replaces original in response to pointer

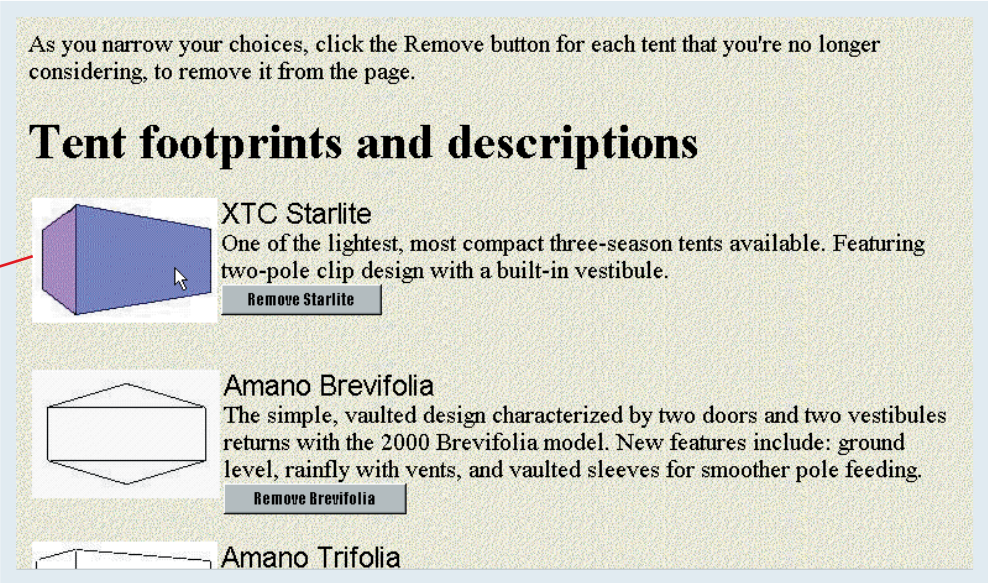



TABLE K-2: List item IDs and graphic filenames

list item	id	color graphic name (onMouseOver)	black and white graphic name (onMouseOut)
1	star	starcolor.jpg	starlite.jpg
2	brev	brevcolor.jpg	brevifolia.jpg
3	tri	trifolia.jpg	tricolor.jpg
4	hillside	hillsidecolor.jpg	hillside.jpg
5	hilltop	hilltopcolor.jpg	hilltop.jpg
6	peak	peakcolor.jpg	peak.jpg
7	summit	summitcolor.jpg	summit.jpg

# Binding Data

DHTML's dynamic content tools offer specialized features for working with tables in your Web pages. One of the most powerful is dynamic table generation, first introduced in Internet Explorer 4. Instead of creating a table using a tag for each element, you can simply create the headers, then add code to reference data located in an external file. Linking an external database with a Web page is known as **data binding**. When the page loads, the browser creates the table at run time. Because the table is re-created each time a user opens the page, you can change the contents of the external data source without changing the Web page code.  Because it's helpful for tent shoppers to be able to compare the details of different models, such as area and weight, Lydia decides to add a tent data table to the Web page. The sales department has provided a text file containing the appropriate information. Lydia binds the file to her Web page to create a dynamic table.

## Steps 1234

1. Open the file **HTML K-6.htm** in your text editor, then save it as a text document with the filename **Tent comparison table.htm**

2. Scroll to near the end of the code until the `<OBJECT>` tags and list of tent descriptions appears in the document window

Figure K-13 shows the code including the `OBJECT` tags. These tags, which Lydia entered earlier, set up the external file containing the data for her table as a Web page object. The `CLASSID` attribute calls the Internet Explorer routine for dynamic table generation to format the linked data. The `PARAM` tags within the beginning and ending `OBJECT` tags denote parameters for this object. The `DataURL` parameter identifies the name of the external file to be bound, which is named `tents.txt`. The `True` value for the `UseHeader` attribute specifies that the data in the external file includes a row of information identifying the contents of each column.

3. Select the text **[replace with opening TABLE tag]**, press **[Delete]**, and type **`<TABLE BORDER="1" ID="element1" DATASRC="#tentlist">`**

The `TABLE` tag formats the code that follows as rows in a table. The `DATASRC` attribute refers to the preceding object, named "tentlist." The number sign indicates that the source is an object in the same Web page.

4. Scroll down, select the text **[replace with closing TABLE tag]**, press **[Delete]**, then type **`</TABLE>`**

The rows within the `TABLE` tags contain row header display information and links to the columns in the external source. The `DATAFLD` attribute in each `DIV` tag names the column header in the external file that marks the column to be associated with the tag. Notice that below the closing `TABLE` tag, Lydia has inserted a script to display extra information for users not running IE4. Because these browsers will not display the bound data, Lydia provides another method for them to obtain the table information.

5. Check your document for errors, make changes as necessary, then save **Tent comparison table.htm** as a text document

6. Open **Tent comparison table.htm** in your browser, then scroll to the bottom of the page  
The tent comparison information from the bound data file appears in a table, as shown in Figure K-14. The sales department can add, remove, or edit lines from the external file, and the Web page table will automatically reflect the most current information each time the Web page is loaded.



**FIGURE K-13: OBJECT tags in Web page source**

Code to  
format  
imported  
data

```
<OBJECT ID="tentlist" CLASSID="clsid:333C7BC4-460F-11D0-BC04-0080C7055A83">
  <PARAM NAME="DataURL" VALUE="tents.txt">
  <PARAM NAME="UseHeader" VALUE="True">
</OBJECT>

[replace with opening TABLE tag]

<THEAD>
<TR>
<TD><B><DIV ID=tent>Tent</DIV></B></TD>
<TD><B><DIV ID=catno>Catalog number</DIV></B></TD>
<TD><B><DIV ID=area>Area (sq ft)</DIV></B></TD>
<TD><B><DIV ID=vest>Vestibule (sq ft)</DIV></B></TD>
<TD><B><DIV ID=desc>Description</DIV></B></TD>
<TD><B><DIV ID=cap>Capacity</DIV></B></TD>
<TD><B><DIV ID=weight>Weight</DIV></B></TD>
<TD><B><DIV ID=price>Price</DIV></B></TD>
</TR>
</THEAD>
<TBODY>
<TR>
<TD><DIV DATAFLD="tent"></DIV></TD>
<TD><DIV DATAFLD="catno"></DIV></TD>
<TD><DIV DATAFLD="area"></DIV></TD>
```

**FIGURE K-14: Tent comparison table**

Browser-  
generated  
table based  
on external  
data source

Tent	Catalog number	Area (sq ft)	Vestibule (sq ft)	Description	Capacity	Weight	Price
XTC Starlite	BR-370	34	10	Staked	1 person	4 lbs. 3 oz.	\$150
Amano Brevifolia	BT-356	38.5	19.6	Freestanding	2 people	5 lbs. 8 oz.	\$215
Amano Trifolia	BT-358	49	25.7	Freestanding	2 people	7 lbs.	\$250
Vista Hillside	BZ-339	32	15.3	Staked	1 person	4 lbs.	\$120
Vista Hilltop	BZ-367	37.5	19.5	Staked	1 person	5 lbs. 3 oz.	\$160
Vista Peak	BZ-323	42.5	24.4	Freestanding	2 people	6 lbs. 3 oz.	\$210
Vista Summit	BZ-334	51.5	28	Freestanding	2 people	7 lbs. 10 oz.	\$275

For more information on Nomad Ltd outdoor supplies, please email our [sales department](#)

# Manipulating Bound Data Dynamically

In addition to dynamic table creation, Internet Explorer 4 introduced other cutting-edge tools for working with tables in Web pages. Perhaps one of the most useful is dynamic sorting, which enables users to sort the data in a table simply by clicking the relevant column heading. To allow users to compare tent statistics based on the most important categories, Lydia adds a script that sorts the tent information on a given column when a user clicks that column heading.

## Steps 1234

1. Open the file **HTML K-7.htm** in your text editor, then save it as a text document with the filename **Tent sortable comparison table.htm**

2. Scroll down to the script beneath the table code near the bottom of the page until function `tentClick()` { is visible

Notice that Lydia has already entered scripts to sort the table. She created a separate script for each column. Each script sorts the table by the contents of that column using the `tentlist.Sort=` command, and then regenerates the table to show the sort, with `tentlist.Reset()`. Accompanying each script is a line of code triggering the script in response to the `onclick` event for the given column header.

3. Scroll to the bottom of the page, select the text **[replace with price script]**, then press **[Delete]**

4. Type the following script, pressing **[Enter]** at the end of each line:

```
function priceClick() {  
    tentlist.Sort="price";  
    tentlist.Reset();  
}  
price.onclick=priceClick;
```

Figure K-15 shows the completed Web page containing the script.

5. Check the script you entered for errors, then save **Tent sortable comparison table.htm** as a text document

6. Open **Tent sortable comparison table.htm** in your Web browser, then scroll to the bottom of the page

The tent comparison table displays in its default order. Notice that the Vestibule (sq ft) column is not displayed in any particular order.

7. Click the **Vestibule (sq ft)** column heading, then scroll down to see the regenerated table

The table disappears, then regenerates to show the records in ascending order by vestibule area, as shown in Figure K-16.

8. Click the **Price** column heading, then scroll down

The table displays the records in order by price, using the script you entered.

9. Close the Web browser and text editor



FIGURE K-15: Web page containing price-sorting script

Script for sorting table on the price column

**price** Column ID for column to be sorted

```

tentlist.Sort="weight";
tentlist.Reset();
}

weight.onclick=weightClick;

function priceClick() {
    tentlist.Sort="price";
    tentlist.Reset();
}

price.onclick=priceClick;

if (!IE4) {
    document.write("If your browser does not display the above table,
please email us at the address below for up-to-date tent details and
prices.<BR><BR>")
}
//-->
</SCRIPT>

<DIV>For more information on Nomad Ltd outdoor supplies, please email our <A
HREF="mailto:sales@nomadltd.com">sales department</A></DIV>

</BODY>
</HTML>

```

Sorts column in ascending order by price

Regenerates the table to show the sort

FIGURE K-16: Table sorted on vestibule area column

Tent	Catalog number	Area (sq ft)	Vestibule (sq ft)	Description	Capacity	Weight	Price
XTC Starlite	BR-370	34	10	Staked	1 person	4 lbs. 3 oz.	\$150
Vista Hillside	BZ-339	32	15.3	Staked	1 person	4 lbs.	\$120
Vista Hilltop	BZ-367	37.5	19.5	Staked	1 person	5 lbs. 3 oz.	\$160
Amano Brevifolia	BT-356	38.5	19.6	Freestanding	2 people	5 lbs. 8 oz.	\$215
Vista Peak	BZ-323	42.5	24.4	Freestanding	2 people	6 lbs. 3 oz.	\$210
Amano Trifolia	BT-358	49	25.7	Freestanding	2 people	7 lbs.	\$250
Vista Summit	BZ-334	51.5	28	Freestanding	2 people	7 lbs. 10 oz.	\$275

For more information on Nomad Ltd outdoor supplies, please email our [sales department](mailto:sales@nomadltd.com)

Table sorted in response to click on column head



## Suppressing errors

When creating cross-browser code, you may want to add features to your pages that generate error messages in some browsers. To allow your information to get out to everyone who wants to view it without alarming viewers, you can include a script that keeps error messages from appearing in incompatible

browsers. By setting the value of the object window.onerror to "null", you prevent error windows from opening when scripts have problems completing. Take care not to add error suppression until you have completed and debugged your page because error suppression removes an important debugging aid.



## ► Concepts Review

Label the code segments marked in Figure K-17.

FIGURE K-17

The figure shows a block of HTML code with five red lines and numbers pointing to specific segments:

- 1** points to the text content: `<DIV>One of the lightest, most compact three-season tents available. Featuring two-pole clip design with a built-in vestibule.</DIV>`
- 2** points to the opening `<DIV>` tag of the first container: `<DIV ID="tent1" name="tent">`
- 3** points to the `Remove Starlite` text inside a JavaScript-generated button: `<BUTTON CLASS='button' onClick=tent1.outerHTML='',reCount()>Remove Starlite</BUTTON>`
- 4** points to the `onMouseOver` event handler: `onMouseOver="star.src='starcolor.jpg'"`
- 5** points to the `onMouseOut` event handler: `onMouseOut="star.src='starlite.jpg'"`

The full code shown in the figure is:

```
<DIV ID="tent1" name="tent">

<DIV CLASS="tenthead" onMouseOver="star.src='starcolor.jpg'"
onMouseOut="star.src='starlite.jpg'"><IMG SRC="starlite.jpg" ALIGN="left"
ID="star">XTC Starlite</DIV>
<DIV>One of the lightest, most compact three-season tents available.
Featuring two-pole clip design with a built-in vestibule.</DIV>

<SCRIPT LANGUAGE="javascript">
<!--
if (IE4) {
    document.write("<BUTTON CLASS='button'
onClick=tent1.outerHTML='',reCount()>Remove Starlite</BUTTON>")
}
//-->
</SCRIPT>

<BR><BR><BR>

</DIV>

<DIV ID="tent2" name="tent">

<DIV CLASS="tenthead" onMouseOver="brev.src='brevcolor.jpg'"
onMouseOut="brev.src='brevifolia.jpg'"><IMG SRC="brevifolia.jpg"
ALIGN="left" ID="brev">Amano Brevifolia</DIV>
```

Match each statement with the term that it describes.

- |  |                    |
|--|--------------------|
| 6. DHTML features that make immediate modifications to a page's actual content | a. Data binding    |
| 7. Period when a browser first interprets and displays a Web page              | b. InnerHTML       |
| 8. Associating an external database with a Web page                            | c. Run time        |
| 9. HTML property for replacing an element and the HTML tags enclosing it       | d. OuterHTML       |
| 10. HTML property for replacing an element but leaving its enclosing HTML tags | e. Dynamic content |

Select the best answer from the list of choices.

11. The outerHTML for the code `<DIV>Welcome to the Nomad Ltd home page!</DIV>` is
- `<DIV>Welcome to the Nomad Ltd home page!</DIV>`.
  - `<DIV>Welcome to the Nomad Ltd home page!`
  - Welcome to the Nomad Ltd home page!.
  - Welcome to the Nomad Ltd home page!</DIV>.

**12. A DHTML clock would be an example of**

- a. Deleting content.
- b. Modifying content.
- c. Adding content.
- d. Dynamic table generation

**13. Which HTML tag set do you use to list the properties for a dynamically generated table?**

- a. <TBL>..</TBL>
- b. <TABLE>..</TABLE>
- c. <THEAD>..</THEAD>
- d. <OBJECT>..</OBJECT>

## ► Skills Review

**1. Insert content dynamically.**

- a. Open the file HTML K-8.htm in your text editor, then save it as a text document with the filename Pack count.htm.
- b. Scroll to the bottom of the Web page code, highlight the text [replace with pack count code], then press [Delete].
- c. Type the following code, pressing [Enter] at the end of each line:
 

```
<SCRIPT>
<!--
if (IE4) {
    countHeaders()
    document.write("<H1 ALIGN='center'>This page describes ")
    document.write(totalPacks)
    document.write(" pack models.</H1>")
}
//-->
</SCRIPT>
```
- d. Check your document for errors, make changes as necessary, then save Pack count.htm as a text document.
- e. Open Pack count.htm in your Web browser, then scroll down to the bottom of the page.

**2. Delete content dynamically.**

- a. Open the file HTML K-9.htm, then save it as a text document with the filename Pack delete.htm.
- b. Scroll down below the body text describing the first pack, the Nomad Moonlight, select the text [replace with button code for pack1], then press [Delete].
- c. Type the following code, pressing [Enter] at the end of each line:
 

```
<SCRIPT LANGUAGE="javascript">
<!--
if (IE4) {
```
- d. Press [Tab], then type document.write("<BUTTON CLASS='button' onClick=pack1.outerHTML=">Remove Moonlight</BUTTON>") and press [Enter].
- e. Type } and press [Enter], then enter the two closing SCRIPT tags.
- f. Repeat Steps b through e for the remaining six pack descriptions, substituting the object names and pack names, as listed in Table K-3.

TABLE K-3

description number	substitute for "pack1"	substitute for "Moonlight"
2	pack2	Blue Moon
3	pack3	Harvest Moon
4	pack4	New Moon
5	pack5	Full Moon
6	pack6	Trekker
7	pack7	Long Haul

- g. Check the document for errors, make changes as necessary, then save Pack delete.htm as a text document.
- h. Open Pack delete.htm in your Web browser, then scroll down the Web page until the Nomad Blue Moon pack description appears in the document window.
- i. If you are using Internet Explorer, click the Remove Blue Moon button.

### 3. Modify content dynamically.

- a. Open the file HTML K-10.htm in your text editor, then save it as a text document with the filename Pack update.htm.
- b. Scroll to the bottom of the Web page code, select the text [replace with opening SPAN tag], then press [Delete].
- c. Type `document.write("<SPAN ID='textnum'>")`
- d. Select the text [replace with closing SPAN tag], press [Delete], type `document.write("</SPAN>")`
- e. Check your document for errors, make changes as necessary, then save Pack update.htm as a text document.
- f. Open Pack update.htm in your Web browser, then scroll to the bottom of the page.
- g. If you are using Internet Explorer, click the Remove Long Haul button.
- h. If you are using Internet Explorer, click the Remove Trekker button.

### 4. Incorporate an advanced function.

- a. Open the file HTML K-11.htm in your text editor, then save it as a text document with the filename Pack scroll.htm.
- b. Scroll down until the opening body tag appears in the document window, select the text [replace with text scroll script], then press [Delete].
- c. Type `onload="scrollit('Find all your outdoor supplies at nomadltd.com!');"`
- d. Check your document for errors, make changes as necessary, then save Pack scroll.htm as a text document.
- e. Open Pack scroll.htm in your Web browser and watch the status bar to see the scrolling text that the new function creates. (*Note:* this feature functions in both Internet Explorer and Navigator.)

### 5. Replace graphics dynamically.

- a. Open the file HTML K-12.htm in your text editor, then save it as a text document with the filename Pack color.htm.
- b. Scroll down until the line `<DIV CLASS="packhead"` appears in the document window, select the text [replace with light event handlers], then press [Delete].
- c. Type `onMouseOver="light.src='lightcolor.jpg'" onMouseOut="light.src='moonlight.jpg'"`
- d. Scroll down and select the text [replace with blue event handlers], press [Delete], then type `onMouseOver="blue.src='bluecolor.jpg'" onMouseOut="blue.src='bluemoon.jpg'"`
- e. Repeat Step d for the remaining five list items, using the IDs and graphic files listed in Table K-4.



TABLE K-4

list item	id	color graphic name (onMouseOver)	black and white graphic name (onMouseOut)
3	harvest	harvestcolor.jpg	harvest.jpg
4	newmoon	newcolor.jpg	newmoon.jpg
5	full	fullcolor.jpg	fullmoon.jpg
6	trek	trekcolor.jpg	trekker.jpg
7	long	longcolor.jpg	longhaul.jpg

- f. Check your document for errors, make necessary changes, then save Pack color.htm as a text document.
- g. Open Pack color.htm in your Web browser.
- h. If you are using Internet Explorer 4, scroll down to the list of pack descriptions, then move your mouse pointer over the heading or graphic for the Nomad Moonlight.
- i. Move your mouse pointer off the selected item.

## 6. Bind data.

- a. Open the file HTML K-13.htm in your text editor, then save it as a text document with the filename Pack comparison table.htm.
- b. Scroll to the end of the code for the list of pack descriptions until the <OBJECT> tags appear in the document window.
- c. Select the text [replace with opening TABLE tag], press [Delete], then type <TABLE BORDER="1" ID="elemtb" DATASRC="#packlist">
- d. Scroll down, select the text [replace with closing TABLE tag], press [Delete], then type </TABLE>
- e. Check your document for errors, make necessary changes, then save Pack comparison table.htm as a text document.
- f. Open Pack comparison table.htm in your browser, then scroll to the bottom of the page. (*Note:* remember that you will see the bound data only in IE 4.)

## 7. Manipulate bound data dynamically.

- a. Open the file HTML K-14.htm in your text editor, then save it as a text document with the filename Pack sortable comparison table.htm.
- b. Scroll to the bottom of the page, select the text [replace with price script], then press [Delete].
- c. Type the following script, pressing [Enter] at the end of each line:

```
function priceClick() {
    packlist.Sort="price";
    packlist.Reset();
}
price.onclick=priceClick;
```
- d. Check the script you entered for errors, make necessary changes, then save Pack sortable comparison table.htm as a text document.
- e. Open Pack sortable comparison table.htm in your Web browser, then scroll to the bottom of the page.
- f. If you are using Internet Explorer 4, click the Price column heading, then scroll down to see the regenerated table.
- g. Close the Web browser and text editor.

## ► Independent Challenges

**1.** The owners of the Green House plant store want to allow online ordering on their Web page. On the page listing the Green House plant store's products, you have started adding a check box next to each item. Users can click on a check box next to each item they want to buy. Also, you have begun to add a line that reports the total number of items the user has marked for purchase.

To complete this independent challenge:

- a. Open the file HTML K-15.htm in your text editor, then save it as a text document with the filename "Green House supply purchase.htm".
- b. Scroll down to the end of the page's head section, select the text [replace with countChecks and reCount functions], then press [Delete].
- c. Type the following lines of script, pressing [Enter] at the end of each line:

```
function countChecks() {  
    items = 0;  
    for (var i = 0; i < document.all.length; i++){  
        var el = document.all[i];  
        if (el.checked){  
            items++;  
        }  
    }  
}
```

```
function reCount() {  
    countChecks()  
    textnum.innerHTML=" " + items;  
}
```

- d. Check the script you entered for errors, make changes as necessary, then save Green House supply purchase.htm as a text document. *Hint:* To view a model of this script refer to the lesson "Modifying content dynamically" and the student files associated with that lesson.
- e. Open "Green House supply purchase.htm" in your browser, then, if you are using Internet Explorer 4, click one of the check boxes and observe the value displayed for total number of items marked for purchase.
- f. Check for errors, then use the text editor to make corrections as needed.
- g. Close the browser and text editor.

**2.** You are creating a Web page containing route information for Sandhills Regional Public Transit. You want to allow riders to compare routes that different bus lines follow; and you want to ensure users can remove from the screen lines that do not apply to them.

To complete this independent challenge:

- a. Open the file HTML K-16.htm in your text editor, then save it as a text document with the filename "SRPT route comparison.htm".
- b. Scroll down to the first list item in the body section, delete the text "[replace with rt11 button code]", then type the following code, pressing [Enter] at the end of each line:

```
<SCRIPT LANGUAGE="javascript">  
<!--  
if (IE4) {
```

- c. Press [Tab], type `document.write("<BUTTON CLASS='button' onClick=rt11.outerHTML=">Remove Route 11</BUTTON>")`
- d. Press [Enter], type `}` and press [Enter], then add the two closing script tags.
- e. Repeat Steps 2 through 4 for the remaining four buttons, replacing the text as indicated in Table K-5.

TABLE K-5

route	replace "rt11" with	replace "Route 11" with
12	rt12	Route 12
13	rt13	Route 13
14	rt14	Route 14
15	rt15	Route 15

- f. Check your work, then save SRPT route comparison.htm as a text document.
- g. Open "SRPT route comparison.htm" in the browser, and if you are using Internet Explorer, test the buttons to be sure they work.
- h. Check for errors, use the text editor to make corrections as needed.
- i. Close the browser and text editor.

**3.** The Community Public School Volunteers organization would like to add a page to their Web site listing schools that currently need volunteers. They also want to include contact information for each school so that potential volunteers can get started immediately. The organization maintains a database of the different schools' volunteer needs. They would like their Web page to reflect the current contents of the database.

To complete this independent challenge:

- a. Open the file HTML K-17.htm in your text editor, then save it as a text document with the filename "CPSV volunteer opportunities.htm".
- b. Select the text [replace with opening TABLE tag], press [Enter], then type the following code:  
`<TABLE BORDER="1" ID="element1" DATASRC="#schools">`
- c. Add the following lines of script after the opening script tags and before the sorting function scripts:  

```
if (!IE4) {
    document.write("If your browser does not display the above table, please contact us to find out about volunteer opportunities.")
}
```
- d. Check your work, make changes as necessary, then save CPSV volunteer opportunities.htm as a text document.
- e. Open CPSV volunteer opportunities.htm in the browser, and, if you are using Internet Explorer 4, test the table to verify that the table-generation and table-sorting functions work correctly.
- f. Check for errors, use the text editor to make corrections as needed. *Hint:* To view model code, refer to the lessons "Generating a table dynamically" and "Manipulating table contents dynamically" as well as all student files associated with these files.
- g. Close the browser and text editor.



**4.** By creating more complex scripts, you can adapt dynamic content features for a wide range of applications. To complete this independent challenge, connect to the Internet and find two Web pages that incorporate dynamic content in ways that are different than those you learned in this unit. Because you can write cross-browser code for some dynamic content features, you can complete this exercise using either Internet Explorer or Netscape Navigator. Print a copy of each page and circle the area of the page where the content changes dynamically. On another sheet of paper, briefly describe how the content changes, what triggers the change, and what qualifies the feature you circled as dynamic content.



## ► Visual Workshop

You have created a Web page for Touchstone Booksellers that makes their book-inventory database available online. They have provided you a preliminary text file that lists several books. Your task is to bind the text file to the Web page to create a dynamically generated table. The client has asked that you make the table sortable by users, as Figure K-18 illustrates. Open the file HTML K-18.htm, save it as a text document with the filename Touchstone sortable inventory.htm, then replace the text “[replace with sorting script]” with the necessary script to make the list sortable. You need to create a set of script code for each column in the database, as you did in the lesson “Manipulating table contents dynamically” in this unit. The script for the first column is:

```
function titleClick() {
    booklist.Sort="title";
    booklist.Reset();
}
title.onclick=titleClick;
```

Table K-6 shows the variables to substitute to create the remaining script segments. Remember to include the opening and closing SCRIPT tags.

TABLE K-6

column	replace all four occurrences of “title” with
2	alast
3	afirst
4	year
5	bind
6	copies

FIGURE K-18

**Current Inventory**

The table below lists our inventory. You can click on any column heading to sort the table on that column.

Table records unordered

Title	Author (last name)	Author (first name)	Year	Binding	Copies on hand
Old Path White Clouds	Nhat Hanh				
Walden	Thoreau				
Complete Poems	Bishop				
Where the Wild Things Are	Sendak				
V for Vendetta	Moore				

Table sorted on Author (last name) column by user click on column heading

**Current Inventory**

The table below lists our inventory. You can click on any column heading to sort the table on that column.

Title	Author (last name)	Author (first name)	Year	Binding	Copies on hand
Complete Poems	Bishop	Elizabeth	1991	hardbound	1
V for Vendetta	Moore	Alan	1988	paperback	1
Old Path White Clouds	Nhat Hanh	Thich	1991	paperback	2
Where the Wild Things Are	Sendak	Maurice	1963	both	3
Walden	Thoreau	Henry David	1854	paperback	4